

# **DETECTION OF DDOS ATTACK USING MACHINE LEARNING ALGORITHM**

A Mini Project Report

submitted by

**FATHIMATH SUHARA M A (MES20MCA-2018)**

to

the APJ Abdul Kalam Technological University  
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications



**Department of Computer Applications**

MES College of Engineering  
Kuttippuram, Malappuram - 679 582

February 2022

## DECLARATION

I undersigned hereby declare that the project report **DETECTION OF DDoS ATTACK USING MACHINE LEARNING ALGORITHM**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Dr. GEEVAR C ZACHARIAS, Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:

FATHIMATH SUHARA M A (MES20MCA-2018)

Date:

DEPARTMENT OF COMPUTER APPLICATIONS  
MES COLLEGE OF ENGINEERING, KUTTIPPURAM



CERTIFICATE

This is to certify that the report entitled **DETECTION OF DDoS ATTACK USING MACHINE LEARNING ALGORITHM** is a bonafide record of the Mini Project work carried out by **FATHIMATH SUHARA M A (MES20MCA-2018)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s)

External Supervisor(s)

Head Of The Department

# Acknowledgements

At the very outset I would like to thank the almighty's mercy towards me over the years , I wish to express my sincere thanks to my project guide Ms. Priya J.D Assistant Professor, Department of Master of Computer Applications who guided me for the successful completeness of this project. I also thank her for valuable suggestions , guidance , constant encouragement , boundless corporation , constructive comments and motivation extended to me for completion of this project work. I would express my sincere thanks to my internal guide Dr. Geevar C Zacharias, Assistant Professor, Department of Master of Computer Applications, for immense guidance to complete the project successfully. I would like to express my sincere thanks to all the faculty members of Master of Computer Applications department for their support and valuable suggestion for doing the project work. Last but not least my graceful thanks to my parents , friends and also the persons who supported me directly and indirectly during the project.

**FATHIMATH SUHARA M A (MES20MCA-2018)**

# Abstract

With the increase in usage of networking technology and the Internet, Intrusion detection becomes important and challenging security problem. A number of techniques came into existence to detect the intrusions on the basis of Machine Learning and Deep Learning procedures. This paper will give inspiration to the use of Machine Learning techniques to IP traffic and gives a concise depiction of every one of the Machine Learning strategies. This paper proposes a DDoS attack detection method based on Machine Learning. Machine Learning methods are compared with regard to their accuracy and detection potential to detect different types of intrusions. The experimental results show that the proposed DDoS attack detection method based on machine learning has a good detection rate for the current popular DDoS attack. Future Research includes Machine Learning methods to find the intrusions so as to improve the detection rate, accuracy and to minimize the false positive rate.

# Contents

|                                      |            |
|--------------------------------------|------------|
| <b>Declaration</b>                   | <b>i</b>   |
| <b>Certificate</b>                   | <b>ii</b>  |
| <b>Acknowledgements</b>              | <b>iii</b> |
| <b>Abstract</b>                      | <b>iv</b>  |
| <b>Contents</b>                      | <b>v</b>   |
| <b>List of Figures</b>               | <b>vii</b> |
| <b>1 Introduction</b>                | <b>1</b>   |
| 1.1 Background . . . . .             | 1          |
| 1.1.1 Motivation . . . . .           | 2          |
| 1.2 Objective . . . . .              | 3          |
| 1.3 Report Organization . . . . .    | 3          |
| <b>2 Literature Survey</b>           | <b>4</b>   |
| <b>3 Methodology</b>                 | <b>5</b>   |
| 3.1 Introduction . . . . .           | 5          |
| 3.2 Developing Environment . . . . . | 7          |
| 3.3 Classifiers . . . . .            | 8          |
| 3.3.1 Models . . . . .               | 8          |
| 3.3.2 Project Plan . . . . .         | 12         |
| 3.3.3 User Story . . . . .           | 13         |
| 3.3.4 Product Backlog . . . . .      | 14         |
| 3.3.5 Sprint Backlog . . . . .       | 15         |
| 3.3.6 Sprint Actual . . . . .        | 15         |
| <b>4 Results and Discussions</b>     | <b>18</b>  |
| 4.1 Datasets . . . . .               | 18         |
| 4.2 Results . . . . .                | 18         |

|                       |           |
|-----------------------|-----------|
| <i>CONTENTS</i>       | vi        |
| <b>5 Conclusions</b>  | <b>20</b> |
| <b>References</b>     | <b>21</b> |
| <b>Appendix</b>       | <b>22</b> |
| Source Code . . . . . | 22        |

# List of Figures

|      |   |    |
|------|---|----|
| 1.1  | DDos Attack Diagram . . . . .                     | 2  |
| 3.1  | Dataset Sample . . . . .                          | 6  |
| 3.2  | Diagram of MLP Classifier . . . . .               | 9  |
| 3.3  | The Hypothesis of Logistic Regression . . . . .   | 9  |
| 3.4  | Diagram of KNN Algorithm . . . . .                | 10 |
| 3.5  | Diagram of Decision Tree . . . . .                | 11 |
| 3.6  | Project Plan . . . . .                            | 12 |
| 3.7  | User Story . . . . .                              | 13 |
| 3.8  | Product Backlog . . . . .                         | 14 |
| 3.9  | Sprint Backlog . . . . .                          | 15 |
| 3.10 | Sprint1 Actual . . . . .                          | 16 |
| 3.11 | Sprint2 Actual . . . . .                          | 16 |
| 3.12 | Sprint3 Actual . . . . .                          | 17 |
| 4.1  | Graphical representation of the results . . . . . | 19 |



# Chapter 1

## Introduction

### 1.1 Background

Distributed denial-of-service attack, also known as DDoS attack, is one of the most common network attacks at present. With the rapid development of computer and communication technology, the harm of DDoS attack is becoming more and more serious. Therefore, the research on DDoS attack detection becomes more important. This type of attack takes advantage of the specific capacity limits that apply to any network resources – such as the infrastructure that enables a company’s website. The DDoS attack will send multiple requests to the attacked web resource – with the aim of exceeding the website’s capacity to handle multiple requests and prevent the website from functioning correctly. In view of this, this paper proposes a predict the accuracy of DDoS attack based on machine learning, which includes two steps: feature extraction and accuracy prediction.

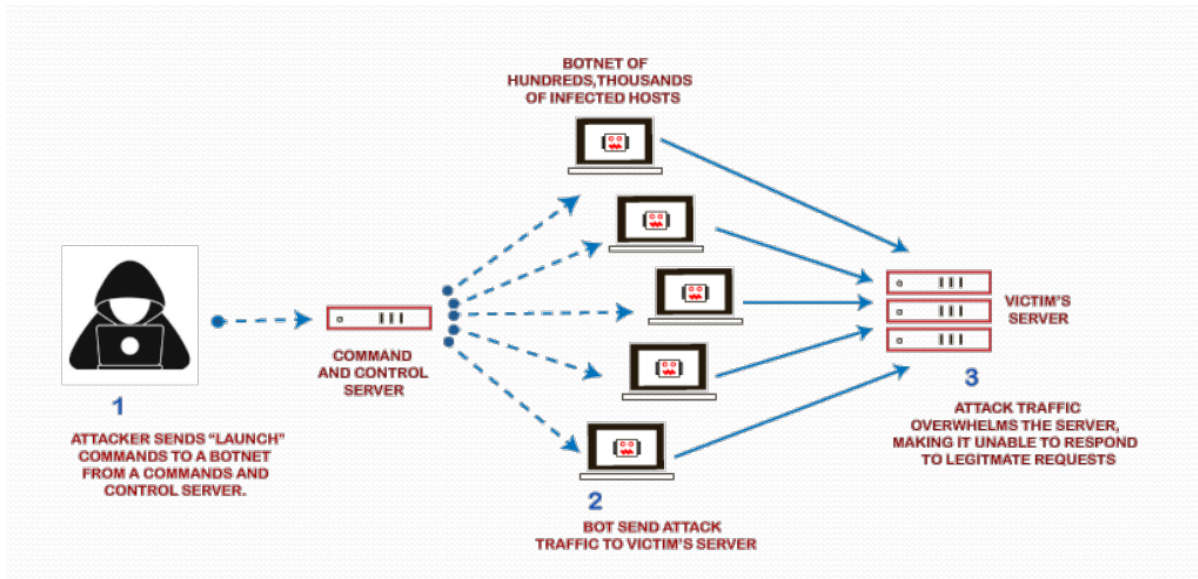


Figure 1.1: DDos Attack Diagram

### 1.1.1 Motivation

Network resources – such as web servers – have a finite limit to the number of requests that they can service simultaneously. In addition to the capacity limit of the server, the channel that connects the server to the Internet will also have a finite bandwidth / capacity. Whenever the number of requests exceeds the capacity limits of any component of the infrastructure, the level of service is likely to suffer in one of the following ways:

- The response to requests will be much slower than normal.
- Some or all users requests may be totally ignored.

Usually, the attacker's ultimate aim is the total prevention of the web resource's normal functioning – a total 'denial of service'. The attacker may also request payment for stopping the attack. In some cases, a DDoS attack may even be an attempt to discredit or damage a competitor's business. In this project, we use several machine learning algorithms to train a model which can be used to detect and classify the type of DDoS attack with greater accuracy. The effect of Denial of Services will be detected by using this machine learning algorithms and the defect can be controlled by taking necessary precautions.

## 1.2 Objective

The aim of the project is to predict the accuracy of the DDoS attacks by using different Machine Learning Algorithms that will be able to achieve the following objectives :

- To analyze data and its parameters to check any redundancy in data values that may affect prediction results.
- To remove all the empty/uncertainty values.
- To detect the DDoS attacks, extract the essential features.
- Use different types of classifiers to compare the model accuracy of detecting the DDoS attacks.
- To be able to generate output within a specified period of time.

## 1.3 Report Organization

The project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results and discussions. Finally Section 5 gives the conclusion.

## Chapter 2

### Literature Survey

William Stallings [1] projected a model with multi class support vector machine (SVM) and chi-square feature selection. Over fitting constant and Radial Basis Function kernel parameter are used for optimization in parameter tuning. They have used NSL-KDD dataset for evaluating the model. They have achieved 98.13 as false alarm rate for the implemented model.

Simon Hansman [2] proposed a new model suitable for use in deep packet inspection based intrusion detection systems using ANN architecture. The proposed ANN architecture achieved an accuracy of 98. Tenfold cross validation. An online exploit and vulnerability repository exploitdb was used to classify the attacks.

Abhishek Pharate , Harsha Bhat , Vaibhav Shilimkar [3] employed a novel approach using multi-layer perceptron (MLP) neural network. Back propagation algorithm is used to train the neural network and minimize the error associated with weights. In this model, they have used KDD99 dataset for training and evaluation. In this model, for feature reduction, PCA algorithm is used. The new method has achieved 91

T. Sree Kala, Dr .A. Christy [4] proposed anomaly based IDS using Long Short Term Memory (LSTM). CIDDS dataset is used for evaluating the model. They have achieved an accuracy of 85.5

# Chapter 3

## Methodology

### 3.1 Introduction

Our objective is to find the best Machine Learning Algorithms for the predict the accuracy of the detection of ddos attack. To achieve this objective, training data have to go through various intermediate processes before giving it to Algorithms. The main parameters on which the performance of the model will be judge are accuracy of the model. First, take the particular Dataset and import it to the Jupyter Notebook. Later by using the machine learning algorithms we can obtain the accuracy of the prediction of DDoS attacks.

#### 1.Data Collection

Data is the prime ingredient of this project, as these data features are various machine learning algorithms. By using these features of the data, Machine Learning Algorithms are trained and models are created. In this proposal, we have taken a ddos attack dataset with 3,11,029 data points and 49 features from Kaggle Website . This data set is divided in the ratio of 70:30 for training and testing of algorithms.

| #   | ▲ Flow ID                            | ▲ Src IP      | # Src Port | ▲ Dst IP     | # Dst Port | # Protocol | 🕒 Timestamp            | # Flow Duration | # Tot Fwd Pkt |
|-----|--------------------------------------|---------------|------------|--------------|------------|------------|------------------------|-----------------|---------------|
|     | 4504-80-6                            |               |            |              |            |            |                        |                 |               |
| 626 | 192.168.4.118-203.73.24.75-4505-80-6 | 192.168.4.118 | 4505       | 203.73.24.75 | 80         | 6          | 12/06/2010 08:34:36 AM | 476078          | 2             |
| 627 | 192.168.4.118-203.73.24.75-4505-80-6 | 192.168.4.118 | 4505       | 203.73.24.75 | 80         | 6          | 12/06/2010 08:34:37 AM | 151             | 2             |
| 628 | 192.168.4.118-203.73.24.75-4506-80-6 | 192.168.4.118 | 4506       | 203.73.24.75 | 80         | 6          | 12/06/2010 08:34:37 AM | 472507          | 2             |
| 629 | 192.168.4.118-203.73.24.75-4506-80-6 | 192.168.4.118 | 4506       | 203.73.24.75 | 80         | 6          | 12/06/2010 08:34:37 AM | 138             | 1             |
| 630 | 192.168.4.118-203.73.24.75-4507-80-6 | 192.168.4.118 | 4507       | 203.73.24.75 | 80         | 6          | 12/06/2010 08:34:37 AM | 469596          | 2             |
| 631 | 192.168.4.118-203.73.24.75-4507-80-6 | 192.168.4.118 | 4507       | 203.73.24.75 | 80         | 6          | 12/06/2010 08:34:37 AM | 1978            | 1             |
| 632 | 192.168.4.118-203.73.24.75-4508-80-6 | 192.168.4.118 | 4508       | 203.73.24.75 | 80         | 6          | 12/06/2010 08:34:37 AM | 468629          | 2             |

Figure 3.1: Dataset Sample

## 2.Data Cleaning

Data set is a chunk of data that is in raw form. It may contain certain symbols like digits, special characters, blank lines, and data without any label. These symbols should be removed as it is of no importance and it can also affect the performance of the model in an adverse manner.

### **3. Data Preprocessing**

After data cleaning, data is now free of unwanted symbols. This data should be converted into the form which can be used for extracting the features easily. It eliminates the null values.

### **4. Feature Selection**

Feature Selection is the process where we select those features which contribute most to your prediction variable or output. Certain features are present in ddos attack, we have to extract them and accordingly, our classifier is trained to predict the ddos attack. For analysis and classification problems we have used eight features including "duration", "service", "src\_bytes", "wrong\_fragment", "count", "urgent", "num\_compromised", "srv\_count". 'duration' is the time gap between sending a request from source port and receiving the request to the destination port, 'service' is the protocol type of the network, 'src\_bytes' is the how much bytes of data are transferred, 'wrong\_fragment', is how data packets or requests are failed, 'count' is the number of the failure data packets, 'urgent' is the state of the data packet, 'num\_compromised' is the details of the attacked data packets and 'srv\_count' is the number of affected data packets.

## **3.2 Developing Environment**

### **Hardware Specification :**

- \* Processor : Intel Pentium Core i3 and above
- \* Primary Memory : 4 GB RAM and above
- \* Storage : 500 GB hard disk and above
- \* Display : VGA Color Monitor
- \* Key Board : Windows compatible
- \* Mouse : Windows compatible

### **Software Specification :**

- \* Language : Python
- \* Front end : Python Django
- \* Back end : SQLite

- \* Operating system : windows 7 and above
- \* IDE : Jupyter Notebook(Anaconda)
- \* Classifier : KNN Algorithm, Logistic Regression Algorithm, Decision Tree Algorithm and MLP Classifier
- \* Dataset : DDoS Dataset from Kaggle Website

### 3.3 Classifiers

In our model, we used four types of machine learning algorithms and for the implementation work, we use the Jupyter notebook platform with the assistance of Python programmable language. The classification models that we implemented using the above-mentioned dataset are Logistic Regression, KNN Algorithm, Decision Tree and MLP Classifier. This algorithms are good for various classifications and that they got their own properties and performance for supporting different datasets.

#### 3.3.1 Models

##### 1. MLP Classifier

The multilayer perceptron (MLP) is a feedforward artificial neural network model that maps input data sets to a set of appropriate outputs. An MLP consists of multiple layers and each layer is fully connected to the following one. The nodes of the layers are neurons with nonlinear activation functions, except for the nodes of the input layer. Between the input and the output layer there may be one or more nonlinear hidden layers.



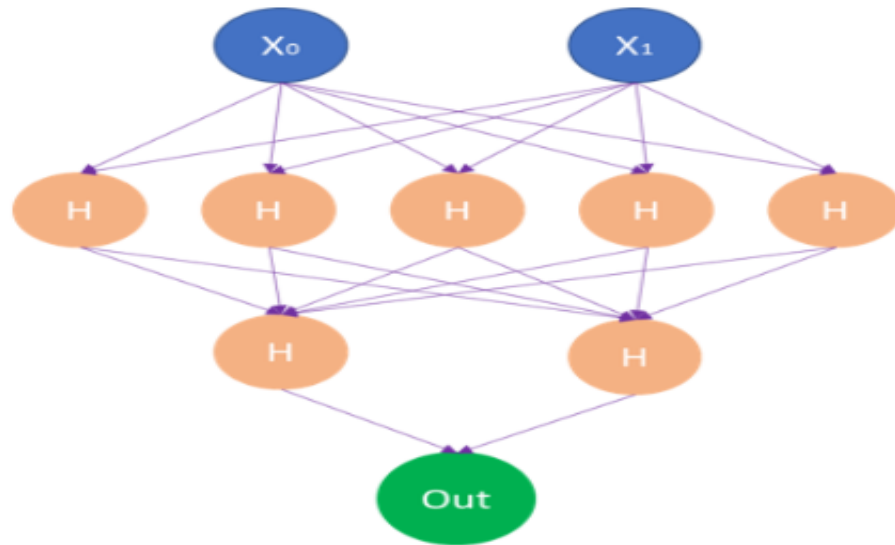


Figure 3.2: Diagram of MLP Classifier

## 2. Logistic Regression Algorithm

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability. We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function, this cost function can be defined as the ‘Sigmoid function’ or also known as the ‘logistic function’ instead of a linear function. The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

$$h\theta(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

Figure 3.3: The Hypothesis of Logistic Regression

### 3. KNN Algorithm

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

#### Why do we need a K-NN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point  $x_1$ , so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

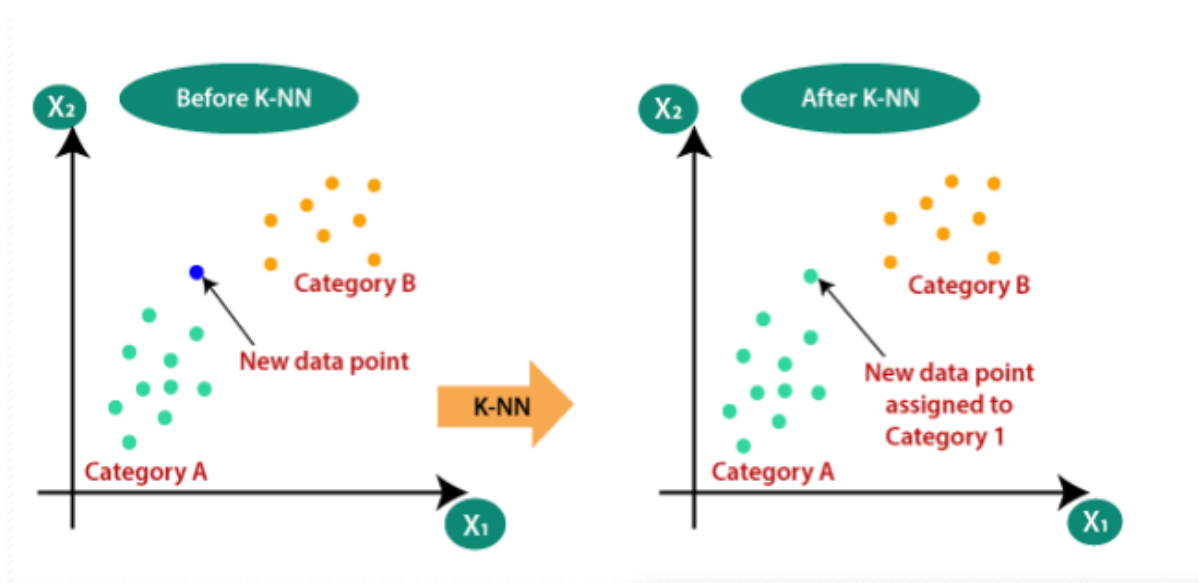


Figure 3.4: Diagram of KNN Algorithm

#### 4. Decision Tree Classification Algorithm

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question, and based on the answer (Yes/No), it further splits the tree into subtrees. Below diagram explains the general structure of a decision tree:

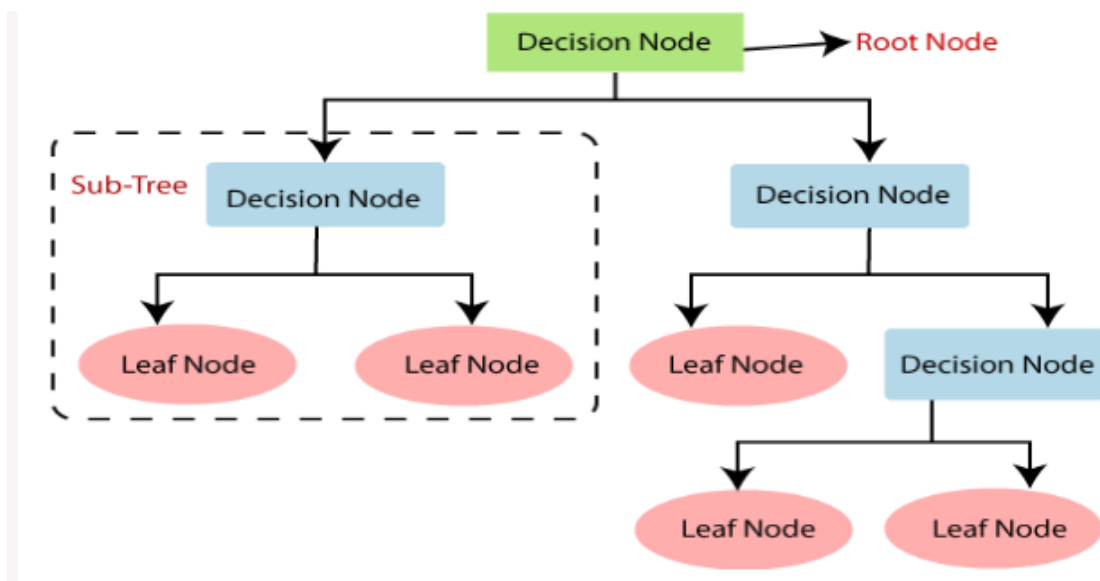


Figure 3.5: Diagram of Decision Tree

### Why use Decision Trees?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- \* Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- \* The logic behind the decision tree can be easily understood because it shows a tree-like structure.

### 3.3.2 Project Plan

A project plan that has a series of tasks laid out for the entire project, listing task duration, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it. Project plan is given below figure. The project has four sprints.

| User Story ID | Task Name | Start Date | End Date   | Days | Status    |
|---------------|-----------|------------|------------|------|-----------|
| 1             | Sprint 1  | 27/12/2021 | 27/12/2021 | 2    | Completed |
| 2             |           | 28/12/2021 | 28/12/2021 |      | Completed |
| 3             | Sprint 2  | 15/01/2022 | 16/01/2022 | 4    | Completed |
| 4             |           | 22/01/2022 | 22/01/2022 |      | Completed |
| 5             |           | 23/01/2022 | 23/01/2022 |      | Completed |
| 6             |           | 26/01/2022 | 29/01/2022 |      | Completed |
| 7             | Sprint 3  | 05/02/2022 | 06/02/2022 | 4    | Completed |
| 8             |           | 12/02/2022 | 12/02/2022 |      | Completed |

Figure 3.6: Project Plan

### 3.3.3 User Story

A key component of agile software development is putting people first, and user-stories put actual end users at the center of the conversation. Stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building what they're building and what value it creates. A user story is a tool used in agile software development to capture a description of a software feature from an end user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work which drives collaboration, creativity, and a better product overall. The user story of system is given in the below figure.

| User Story ID | As a type of User | I Want To<br><Perform Some Task>                         | So that I can<br>< Achieve Some Goal>           |
|---------------|-------------------|--|---|
| 1             | User              | Collection of datasets                                   | DDoS Dataset                                    |
| 2             | User              | Preprocessing of collected data (null value elimination) | Cleaned final dataset                           |
| 3             | User              | Collect IP Address,Port Number                           | Get IP Address and Port number                  |
| 4             | User              | Collect incoming packets                                 | Get incoming packets                            |
| 5             | User              | Count the number of accesses or IP Address               | Get the number of accesses                      |
| 6             | User              | Compare with dataset                                     | Compare the number of access with the threshold |
| 7             | User              | Modelling the data                                       | Apply ML Algorithms                             |
| 8             | User              | Output Generation  | Detecting DDoS Attack Or No Attack              |

Figure 3.7: User Story

### 3.3.4 Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome. The product backlog is the single authoritative source for things that a team works on. That means that nothing gets done that is not on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment. It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. The Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories. The product backlog of the system is given below figure.

| User story ID | Priority<br><High/Medium/Low> | Size<br>(Hours) | Sprint<br><#> | Status<br><Planned/In<br>progress/Completed> | Release Date               | Release Goal                               |
|---------------|-------------------------------|-----------------|---------------|--|----------------------------|--|
| 1             | Medium                        | 2               | 1             | Completed                                    | 27/12/2021                 | Collection of datasets from Kaggle website |
| 2             | Medium                        | 2               |               | Completed                                    | 28/12/2021                 | Preprocessing of collected data            |
| 3             | Medium                        | 5               | 2             | Completed                                    | 15/01/2022<br>-16/01/2022  | Collect IP Address, Port No                |
| 4             | High                          | 10              |               | Completed                                    | 22/01/2022                 | Collect Incoming Data packets              |
| 5             | Medium                        | 5               |               | Completed                                    | 23/01/2022                 | Count number of accesses or IP Address     |
| 6             | High                          | 10              |               | Completed                                    | 26/01/2022<br>- 29/01/2022 | Compare with dataset                       |
| 7             | High                          | 6               | 3             | Completed                                    | 06/02/2022                 | Apply ML Algorithms                        |
| 8             | High                          | 5               |               | Completed                                    | 12/02/2022                 | Output Generation                          |

Figure 3.8: Product Backlog

### 3.3.5 Sprint Backlog

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story. Most teams also estimate how many hours each task will take someone on the team to complete.

| Backlog Item             | Status& Completion | Original Estimate in hours | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 | Day 8 | Day 9 | Day 10 | Day 11 |
|--------------------------|--------------------|----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|
| User Story#1             |                    | Hours                      | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours  | Hours  |
| Dataset Collection       | 27/12/2021         | 5                          | 3     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      |
| Preprocessing            | 28/12/2021         | 5                          | 0     | 2     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      |
| User Story #2,3,4,5,6    |                    |                            |       |       |       |       |       |       |       |       |       |        |        |
| Visualization & Training | 23/01/2022         | 10                         | 0     | 0     | 4     | 4     | 2     | 0     | 0     | 0     | 0     | 0      | 0      |
| User Story #7            |                    |                            |       |       |       |       |       |       |       |       |       |        |        |
| UI Designing             | 29/01/2022         | 12                         | 0     | 0     | 0     | 0     | 0     | 4     | 4     | 4     | 0     | 0      | 0      |
| User Story #8            |                    |                            |       |       |       |       |       |       |       |       |       |        |        |
| Testing                  | 12/02/2022         | 18                         | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 6     | 6      | 6      |
| Total                    |                    | 50                         | 5     | 5     | 4     | 4     | 2     | 4     | 4     | 4     | 6     | 6      | 6      |

Figure 3.9: Sprint Backlog

### 3.3.6 Sprint Actual

Actual sprint backlog is what adequate sprint planning is actually done by project team there may or may not be difference in planned sprint backlog. The detailed sprint backlog (Actual) is given below.

| Backlog Item             | Status& Completion | Original Estimate in hours | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 | Day 8 | Day 9 | Day 10 | Day 11 | Completed <Y/N> |
|--------------------------|--------------------|----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|-----------------|
| User Story#1             |                    | Hours                      | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours  | Hours  |                 |
| Data Collection          | 27/12/2021         | 5                          | 5     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | Y               |
| Preprocessing            | 28/12/2021         | 5                          | 0     | 5     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | Y               |
| User Story #2,3,4,5,6    |                    |                            |       |       |       |       |       |       |       |       |       |        |        |                 |
| Visualization & Training | 23/01/2022         | 10                         | 0     | 0     | 4     | 4     | 2     | 0     | 0     | 0     | 0     | 0      | 0      | N               |
| User Story #7            |                    |                            |       |       |       |       |       |       |       |       |       |        |        |                 |
| UI Designing             | 29/01/2022         | 12                         | 0     | 0     | 0     | 0     | 0     | 4     | 4     | 4     | 0     | 0      | 0      | N               |
| User Story #8            |                    |                            |       |       |       |       |       |       |       |       |       |        |        |                 |
| Testing                  | 12/02/2022         | 18                         | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 6     | 6      | 6      | N               |
| Total                    |                    | 50                         | 5     | 5     | 4     | 4     | 2     | 4     | 4     | 4     | 6     | 6      | 6      | N               |

Figure 3.10: Sprint1 Actual

| Backlog Item             | Status& Completion | Original Estimate in hours | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 | Day 8 | Day 9 | Day 10 | Day 11 | Completed <Y/N> |
|--------------------------|--------------------|----------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|-----------------|
| User Story#1             |                    | Hours                      | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours  | Hours  |                 |
| Data Collection          | 27/12/2021         | 5                          | 5     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | Y               |
| Preprocessing            | 28/12/2021         | 5                          | 0     | 5     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | Y               |
| User Story #2,3,4,5,6    |                    |                            |       |       |       |       |       |       |       |       |       |        |        |                 |
| Visualization & Training | 23/01/2022         | 10                         | 0     | 0     | 4     | 4     | 2     | 0     | 0     | 0     | 0     | 0      | 0      | Y               |
| User Story #7            |                    |                            |       |       |       |       |       |       |       |       |       |        |        |                 |
| UI Designing             | 29/01/2022         | 12                         | 0     | 0     | 0     | 0     | 0     | 4     | 4     | 4     | 0     | 0      | 0      | Y               |
| User Story #8            |                    |                            |       |       |       |       |       |       |       |       |       |        |        |                 |
| Testing                  | 12/02/2022         | 18                         | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 6     | 6      | 6      | N               |
| Total                    |                    | 50                         | 5     | 5     | 4     | 4     | 2     | 4     | 4     | 4     | 6     | 6      | 6      | N               |

Figure 3.11: Sprint2 Actual



| Backlog Item                | Status&<br>Completion | Original<br>Estimate<br>in hours | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 | Day 8 | Day 9 | Day 10 | Day<br>11 | Completed<br><Y/N> |
|-----------------------------|-----------------------|----------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|-----------|--------------------|
| User Story#1                |                       | Hours                            | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours | Hours  | Hours     |                    |
| Data Collection             | 27/12/2021            | 5                                | 5     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0         | Y                  |
| Preprocessing               | 28/12/2021            | 5                                | 0     | 5     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0         | Y                  |
| User Story<br>#2,3,4,5,6    |                       |                                  |       |       |       |       |       |       |       |       |       |        |           |                    |
| Visualization &<br>Training | 23/01/2022            | 10                               | 0     | 0     | 4     | 4     | 2     | 0     | 0     | 0     | 0     | 0      | 0         | Y                  |
| User Story #7               |                       |                                  |       |       |       |       |       |       |       |       |       |        |           |                    |
| UI Designing                | 29/01/2022            | 12                               | 0     | 0     | 0     | 0     | 0     | 4     | 4     | 4     | 0     | 0      | 0         | Y                  |
| User Story #8               |                       |                                  |       |       |       |       |       |       |       |       |       |        |           |                    |
| Testing                     | 12/02/2022            | 18                               | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 6     | 6      | 6         | Y                  |
| Total                       |                       | 50                               | 5     | 5     | 4     | 4     | 2     | 4     | 4     | 4     | 6     | 6      | 6         | Y                  |

Figure 3.12: Sprint3 Actual

## Chapter 4

# Results and Discussions

### 4.1 Datasets

In this project dataset used is DDoS attack dataset which is downloaded from kaggle website. DDoS attack dataset contains a lots of datas.Divide this dataset into training data and test data.We train and create a model using training data from the dataset and we test using test data to find the accuracy for the detection of DDoS attack.

| Dataset      | Origin         | Sample   | Features |
|--------------|----------------|----------|----------|
| DDoS Dataset | Kaggle Website | 3,11,029 | 49       |

### 4.2 Results

In this project,we use several machine learning algorithms to train a model which can be used to detect and classify the type of DDoS attack with greater accuracy.The above figures shows each model give better accuracy for the detection of DDoS attacks.The effect of Denial of Services will be detected by using this machine learning algorithms and the defect can be controlled by taking necessary precautions.

| MLP Classifier | Logistic Regression Algorithm | KNN Algorithm | Decision Tree Algorithm |
|----------------|-------------------------------|---------------|-------------------------|
| 99.9979        | 99.9939                       | 99.9939       | 99.7959                 |

Table 4.1: Accuracy of the Detection

```
In [28]: plt.plot(classifiers,scores)
plt.title("Attack")
plt.ylim(99.5,100)
plt.show()
print(y_pred[2])
df.to_csv("d:\\revised_kddcup_dataset.csv")
```

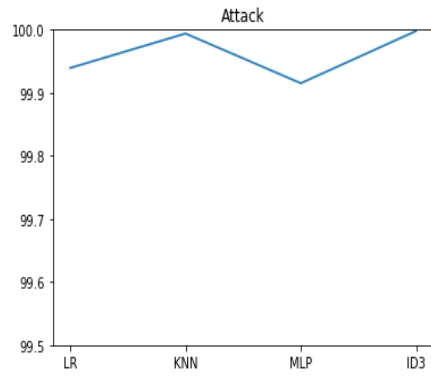


Figure 4.1: Graphical representation of the results

The experimental results show that the proposed DDoS attack prediction method based on machine learning has a good detection rate for the current popular DDoS attack.

## Chapter 5

### Conclusions

A review has been made of the significant works in the fields of machine learning that are used to find the accuracy for the detection of DDoS attack. All these works have emphasized distinct machine learning strategies utilized, informational index or datasets utilized, assessment measurements for every one of the systems utilized. To decide the powerful approach a few criteria must be considered. The criterion included classification time, training time, detection rate and accuracy. It is hard to distinguish a superior methodology of ML strategies dependent on just one factor like accuracy. On the off chance that the ML strategies looked at dependent on precision or accuracy.

## References

- [1] **William Stallings**, “Cryptography and Network Security”, 5 th edition, Pearson Education.
- [2] **Simon Hansman**. “A Taxonomy of Network and Computer Attack Methodologies” 2003.
- [3] **Abhishek Pharate , Harsha Bhat , Vaibhav Shilimkar** “Classification of Intrusion Detection System” IJCS Volume 118 – No. 7, May 2015.
- [4] **T. Sree Kala, Dr .A. Christy** “A Survey and Analysis of Machine Learning Algorithms for Intrusion Detection System” Jour of Adv Research in Dynamical and Control Systems, 04-Special Issue, June 2017.

# Appendix

## Source Code

---

```

#!pip install lib_ddos_simulator
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
colnames =
    ["duration", "protocol_type", "service", "flag", "src_bytes", "dst_bytes", "land", "wrong_fragment", "urgent", "hot", "num_failed_logins", "logged_in"]
len(colnames)
df =
    pd.read_csv("C:\\Users\\fathi\\Downloads\\DDoS_Detection-master-20220205T052511Z-001\\DDoS_Detection-master\\dataset\\corrected.csv", header=0)
df.head(100)
df.shape
#extracting the icmp packets from our dataset
icmp_df = df[df.loc[:, "protocol_type"] == "icmp"]
#none of the values in dataset are null
icmp_df.isnull().sum()
icmp_df.head()
#I will be extracting all the important features as a "priori" for preprocessing
features = ["duration", "service", "src_bytes", "wrong_fragment", "count", "urgent", "num_compromised", "srv_count"]
target = "result"
X = icmp_df.loc[:, features]
y = icmp_df.loc[:, target]
classes = np.unique(y)
print(classes)
#replacing all classes of attack with 1 and normal result with 0 in our icmp_df
for i in range(len(classes)):
    if i == 2:
        icmp_df = icmp_df.replace(classes[i], 0)
    else:
        icmp_df = icmp_df.replace(classes[i], 1)

#turning the service attribute to categorical values
icmp_df=icmp_df.replace("eco_i",-0.1)
icmp_df=icmp_df.replace("ecr_i",0.0)
icmp_df=icmp_df.replace("tim_i",0.1)
icmp_df=icmp_df.replace("urp_i",0.2)
y = icmp_df.loc[:,target]
#I selected certain features but I will have to find some covariance between them so I will plot a covariance heatmap
sns.heatmap(X.corr(), annot=True, cmap="RdBu")
plt.plot()
#the data as seen is highly uncorrelated as most of it is one valued such as the duration one.
X = icmp_df.loc[:, features]
y = icmp_df.loc[:, target]
X.head(100)

```

---

## Appendix

---

```
print(list(X.loc[629,:])) #7 input features
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42, test_size=0.3)
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
models = [LogisticRegression(), KNeighborsClassifier(n_neighbors=3), MLPClassifier(alpha=0.005), DecisionTreeClassifier()]
classifiers = ["LR", "KNN", "MLP", "ID3"]
scores = []
for model in models:
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    score = accuracy_score(y_test, y_pred)*100
    scores.append(score)
    print("Accuracy of the model is: ", score)
    conf_matrix = confusion_matrix(y_test, y_pred)
    report = classification_report(y_test, y_pred)
    print("Confusion Matrix:\n", conf_matrix)
    print("Report:\n", report)
    print("\n=====**=====")
    plt.plot(classifiers, scores)
plt.title("Attack")
plt.ylim(99.5, 100)
plt.show()
print(y_pred[2])
df.to_csv("d:\\revised_kddcup_dataset.csv")
```

---